PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04L 9/08 | A1

(11) International Publication Number: **WO 99/49613**

(43) International Publication Date: 30 September 1999 (30.09.99)

(21) International Application Number: PCT/US99/03665

(22) International Filing Date: 19 February 1999 (19.02.99)

(30) Priority Data:
60/075,330 20 February 1998 (20.02.98) US

(71) Applicant *(for all designated States except US)*: FORTRESS TECHNOLOGIES, INC. [US/US]; Suite 650, 2701 North Rocky Point Drive, Tampa, FL 33607 (US).

(71)(72) Applicants and Inventors: FRIEDMAN, Aharon [US/US]; 2717 Seville Boulevard, Clearwater, FL 34624 (US). BOZOKI, Eva [US/US]; 6 Lantern Court, Stony Point, NY 11790 (US).

(74) Agent: GOLDMAN, Gregg, I.; Proskauer Rose LLP, 1585 Broadway, New York, NY 10036 (US).

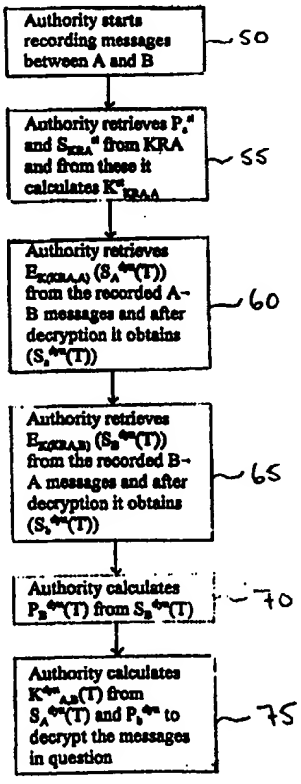
(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, HR, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: CRYPTOGRAPHIC KEY-RECOVERY MECHANISM

(57) Abstract

Nodes I, I=1, N are communicating with each other encrypted. They each have static private (S_i) and public (P_i) keys, which never change and dynamic private (S_i^{dyn}) and public (P_i^{dyn}) keys, which are functions of time (t). A key recovery authority (KRA) also has static private (S_{KRA}) and public (P_{KRA}) keys, which never change. The KRA exchanges static public keys with each of the nodes, thus develops a static common key (session key), $K_{KRA,I}$, with each of them using, for example, the Diffie–Hellman protocol. The KRA maintains a list of the static public keys of all nodes. Thus, the (static) session key with any of the nodes can be "recovered" at any time. When two nodes, say i and j, exchange their dynamic public keys (encrypted with their static session key $K^{st}_{ij}(t)$), then each one attaches its dynamic secret key, encrypted with the static session key between it and the KRA. A time stamp is also included. With knowledge of the session key, $K_{KRA,I}$, which can be recovered from the KRA, the dynamic private keys of each node, $S_i^{dyn}(t)$, can be recovered (and $P_i^{dyn}(t)$ calculated) from a recording of any session (70). From $S_i^{dyn}(t)$ and $P_j^{dyn}(t)$ one can calculate the dynamic session key between the two nodes ($K_{I,J}^{dyn}(t)$) (75). However, all other parties are still protected since their dynamic public keys are exchanged encrypted. Note that all nodes are still protected, and their session concealed, because their private keys are encrypted.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

CRYPTOGRAPHIC KEY-RECOVERY MECHANISM

Field of the Invention

The present invention is directed to cryptography and, more particularly, to a key escrow and key recovery method for use with a cryptography system using static (permanent) and dynamic (changing over time) cryptographic keys.

Background of the Invention

Cryptography has become essential to the acceptance of electronic commerce and sensitive electronic communications over a network. For example, secure digital signatures and verification methods provide high assurance that a party is who it represents itself to be in order to prevent unauthorized users and eavesdropping. This assurance is vital to the general acceptance of, for example, commerce over the Internet, the use of electronic money, cellular communications, and remote computer login procedures. Typically, certain well-known cryptographic methods are used to encrypt information in a manner that is very difficult to decrypt without certain secret information, thus making these signatures and verifications secure. One type of cryptographic method which is commonly used is public key cryptography.

Eavesdropping in a network can be thwarted through the use of a message encryption technique. A message encryption technique employs an encipherment function which utilizes a number referred to as a session key to encipher data (i.e., message content). Only the pair of hosts in communication with each other have knowledge of the session key,

so that only the proper hosts, as paired on a particular conversation, can encrypt and decrypt digital signals. Two examples of encipherment functions are the National Bureau of Standards Data Encryption Standard (DES) (see e.g., National Bureau of Standards, "Data Encryption Standard", FIPS-PUB-45, 1977) and the more recent Fast Encipherment Algorithm (FEAL)(see e.g., Shimizu and S. Miyaguchi, "FEAL-Fast Data Encipherment Algorithm," Systems and Computers in Japan, Vol. 19, No. 7, 1988 and S. Miyaguchi, "The FEAL Cipher Family", Proceedings of CRYPTO '90, Santa Barbara, Calif., Aug., 1990). Another encipherment function is known as IDEA. One way to use an encipherment function is the electronic codebook technique. In this technique a plain text message m is encrypted to produce the cipher text message c using the encipherment function f by the formula c=f(m,sk) where sk is a session key. The message c can only be decrypted with the knowledge of the session key sk to obtain the plain text message m=f(c,sk).

Session key agreement between two communications hosts may be achieved using public key cryptography. (See e.g., U.S. Patent Nos. 5,222,140, 5,299,263).

Before discussing public key cryptographic techniques, it is useful to provide some background information. Most practical modern cryptography is based on two notorious mathematical problems believed (but not proven) to be hard (i.e., not solvable in polynomial time, on the average). The two problems are known as Factorization and Discrete-Log. The Factorization problem is defined as follows:

Input: N, where N=pq where p and q are large prime numbers

Output: p and/or q.

The Discrete-Log problem is defined as follows:

Input: P,g,y, where $y \equiv g^x \bmod P$, and P is a large prime number

Output: x.

(The Discrete-Log problem can be similarly defined with a composite modulus N=pq).

Based on the Factorization and Discrete-Log problems, some other problems have been defined which correspond to the cracking problems of a cryptographic system.

One system of such a problem which has previously been exploited in cryptography (see, e.g., H.C. Williams, "A Modification of RSA Public-Key Encryption", IEEE Transactions on Information Theory, Vol. IT-26, No. Nov. 6, 1980) is the Modular Square Root problem, which is defined as follows:

Input: N,y, where $y \equiv x^2 \bmod N$, and N=pg, where p and q are large primes

Output: x.

Calculating square roots is easy if p and q are known but hard if p and q are not known. When N is composed of two primes, there are in general four square roots mod N. As used herein, $z \equiv \sqrt{x} \bmod N$ is defined to mean that x is the smallest integer whereby $z^2 \equiv x \bmod N$.

Another problem is known as the Composite Diffie-Hellman (CDH) problem, which is defined as follows:

Input: N, g, $g^x \bmod N$, $g^y \bmod N$, where $N \equiv pq$ and p and q are large primes.

Output: $g^{xy} \bmod N$.

It has been proven mathematically, that the Modular Square Root and Composite Diffie-Hellman problems are equally difficult to solve as the above-mentioned factorization problem (see, e.g., M.O. Rabin, "Digitalized Signatures and Public Key Functions as Intractable as Factorization", MIT Laboratory for Computer Science, TR 212, Jan. 1979;

Z. Shmuely, "Composite Diffie-Hellman Public Key Generating Schemes Are Hard To Break", Computer Science Department of Technion, Israel, TR 356, Feb. 1985; and K.S. McCurley, "A Key Distribution System Equivalent to Factoring:, Journal of Cryptology, Vol. 1, No. 2, 1988, pp. 95-105).

In a typical public-key cryptographic system, each user i has a public key P_i (e.g., a modulus N) and a secret key S_i (e.g., the factors p and q). A message to user i is encrypted using a public operation which makes use of the public key known to everybody (e.g., squaring a number mod N). However, this message is decrypted using a secret operation (e.g., square root mod N) which makes use of the secret key (e.g., the factors p and q).

Public key cryptographic techniques may be used for authentication. Authentication is a (theoretically) fool-proof technique for a party to verify that a party contacting it is the party is asserts to be. For example, a confidential network may require that a party authenticate itself before gaining access to the network.

Fig. 1A is a block diagram of a typical cryptography device 100 that may be utilized in the present invention. The device 100 has a processor 102 including one or more CPUs 102, a main memory 104, a disk memory 106, an input/output device 108, and a network interface 110. The devices 102-110 are connected to a bus 120 which transfers data, i.e., instructions and information between each of these devices 102-110.

Fig. 1B illustrates a network 150 over which cryptography devices 100 may communicate and which may be utilized in the present invention. Two or more cryptography devices 100, 100' may be connected to a communications network 152, such as a wide area network; which may be the Internet, a telephone network, or leased lines; or a local area network. Each device 100 may include a modem 154 or other network

communication device to send encrypted messages over the communications network 152. A cryptography device 100 may be a gateway to a sub-network 156. That is, the device 100 may be an interface between a wide area network 152 and a local area (sub) network 156.

An example of a public key cryptographic technique which may be performed by the device 100 is the well known Diffie-Hellman key exchange protocol. The Diffie-Hellman protocol conventionally provides a partially secure distribution system utilizing a symmetric crypto-key between two nodes of a local area network (LAN) or wide area network (WAN). In this protocol, both nodes compute their common crypto-key from their own private key, as well as from the other node's public key. The nodes exchange their public keys, but maintain (for security) their computed crypto-key.

For example, assume two nodes wish to communicate with each other via encrypted packet information. Each has their own private and public key, and consequently each pair of nodes will compute a different common secret crypto-key, which in turn will be used in a symmetric algorithm (using, e.g., well-known DES or IDEA algorithms, discussed above). Typically, the private key of each node is changed periodically. This will lead to two Diffie-Hellman key exchanges in each period, since the nodes do not have to be synchronized.

Further, it is known to use two private and two public keys in each node, i.e., one static key, which never changes, and one dynamic key, which is changed periodically (e.g., every 24 hours), in each private and public pair. One can use the static common crypto-key, developed via a Diffie-Hellman key exchange, to encrypt every consecutive dynamic key exchange.

Summary of the Invention

Described is a key escrow and key recovery method suitable for use with cryptography devices, such as the NetFortress™ VPN family of products. These products use four keys, static (permanent) private and public keys and dynamic (changes over time) private and public keys. Briefly, each node shares a permanent session key with a key recovery authority (KRA) and every pair of nodes share a permanent and a dynamic session key with each other. When two nodes initiate communication, the nodes exchange dynamic public keys (encrypted with a static common key shared by the two nodes), each node also sends its dynamic private key encrypted with the session key it shares with the KRA. Because neither node knows the other node's session key with its KRA, it cannot decrypt the dynamic private key. However, a third party having a court order may be able to obtain the node/KRA session key for the two communicating nodes and thus obtain the dynamic private key for each node, permitting it to decrypt messages encrypted with the nodes' dynamic crypto key.

In particular, nodes I, I=1, N are communicating with each other encrypted. They each have static private (S_i) and public (P_i) keys, which never change and dynamic private (S_i^{dyn}) and public (P_i^{dyn}) keys, which are functions of time (t). A key recovery authority (KRA) also has static private (S_{KRA}) and public (P_{KRA}) keys, which never change. The KRA exchanges static public keys with each of the nodes, thus develops a static common key (session key), $K_{KRA,i}$, with each of them using, for example, the Diffie-Hellman protocol. The KRA maintains a list of the static public keys of all nodes. Thus, the (static) session key with any of the nodes can be "recovered" at any time.

When two nodes, say i and j, exchange their dynamic public keys (encrypted with their static session key $K^{st}_{ij}(t)$), then each one attaches its dynamic secret key, encrypted with the static session key between it and the KRA. A time stamp is also included. With the knowledge of the session key, $K_{KRA,i}$, which can be recovered from the KRA, the dynamic private keys of each node, $S_i^{dyn}(t)$, can be recovered (and $P_i^{dyn}(t)$ calculated) from a recording of any session. From $S_i^{dyn}(t)$ and $P_j^{dyn}(t)$ one can calculate the dynamic session key between the two nodes ($K_{i,j}^{dyn}(t)$). However, all other parties are still protected since their dynamic public keys are exchanged encrypted. Note that all nodes are still protected, and their session concealed, because their private keys are encrypted.

Brief Description of the Drawing

The present invention is described with reference to the following figures:

FIG 1A is a block diagram of a typical cryptography device;

FIG 1B illustrates a communications network over which cryptography devices may communicate;

FIG 2 schematically illustrates a VPN that may be used in accordance with the present invention; and

FIG 3 is a flowchart illustrating the steps taken to decrypt messages between two nodes by a third party in accordance with the present invention.

Detailed Description of the Invention

Preliminaries

The term "key recovery" is used herein as a generic term encompassing the various key escrow, trusted third-party, exceptional access, data recovery and key recovery encryption systems. All these key recovery systems share the following essential elements relevant to this invention:

* A mechanism, external to the primary means of encryption and decryption, by which a third party (such as a government law enforcement agency) can obtain covert access to the plaintext of encrypted data.
* the existence of a highly sensitive secret key (or collection of keys) which must be secured for an extended period of time.

In a network similar to network 150 illustrated in FIG 1B, we may assume that illustratively, devices 100 (A) and 100' (B) are each units from the NetFortress™ VPN family of products (VPN-1, VPN-3 or Remote), available from Fortress Technologies, Tampa, Florida, which products use Fortress Technologies' SPS (Secret Packet Shield™) core technology, such as described in U.S. patent number 5,757,924 to Friedman et al. and owned by Fortress Technologies. The contents of this patent are fully incorporated herein by reference. Of course, any cryptography devices may be used, as desired, which are programmed to perform the inventive method described below.

FIG 2 schematically illustrates a VPN 100 (a network security device) that may be used in accordance with the invention. The security device 10 comprises a first interface 0 which is connected to the client host 12. Specifically, the interface 0 is connected to a network interface in the client host 12 via a cable or wire 13. The security device 10

comprises a second interface 1 which is connected to a portion of a network 150, such as the one described in FIG 1B. Illustratively, the interface 1 is connected to an Ethernet so that the interfaces 0,1 are Ethernet interfaces such as SMC Elite Ultra Interfaces.

A CPU 14 is connected to the interfaces 0,1. The CPU is for example an Intel 486 DX 62-66. A static memory 16 (e.g. flash EEPROM) is connected to the CPU 14 and a dynamic memory 18 (e.g. RAM) is connected to the CPU 14. An optional encryption module 20 performs encryption and large number arithmetic operations. The encryption unit may be implemented as a programmable logic array. Alternatively, the encryption module may be omitted and its function may be carried out using a software program which is executed by the CPU 14. The interface 0 is put in a promiscuous mode. In this mode, the interface 0 passes all communications from the client host 12 that is sensed on the cable 13 to the CPU 14. The network connection is via the interface 1 which is set to the same IP address as the client 12. The VPN 100 responds to the Address Resolution Protocol by sending its own (rather than the client's) MAC address. This adds a level of security by blocking attempts to bypass the device 10 using the Ethernet protocol.

The CPU 14 maintains two databases. One database is a static database stored in the Flash ROM 16. This database contains permanent information about secured nodes in the network, i.e., the node IP address, time entered into the database, the nodes permanent public key.

A second database is a dynamic database. The dynamic database contains information about secured and unsecured nodes, i.e., the node IP address, time last updated, a flag indicating whether the node is secured (e.g., has its own network security device), a flag indicating whether the node is in transition (i.e., in the middle of a key exchange), a

pointer to a common secret key with that node. The transition flag has three possible values, 0-not in transition, 1-pending reply from remote host, and 2-pending computation of common key.

The software executed by the CPU 14 has three components: (1) operating system, (2) networking system, (3) key computation algorithms. The operating system and the networking system are both part of a Unix like kennel. The key computation algorithm reside in memory and are signaled into action by the networking system. The operating system can be colorfully described as a lobotomized Linux system with all drives taken out except the RAM, disk and Ethernet interfaces. The networking system is for communication, key exchange, encryption, configuration, etc.

Public key cryptography can be used to negotiate securely a unique common secret key between any two VPN units. Each unit has four keys associated with it: static (remaining the same during the lifetime of the unit, and characteristic to that unit) private, and public keys S_A^{st} and P_A^{st}, and dynamic (changing periodically) private, and public keys, $S_A^{dyn}(t)$ and $P_A^{dyn}(t)$.

KRA is the Key Recovery Authority which stores the static public keys of all VPN units under its jurisdiction: $P_i^{st}(t)$. Note that the KRA will typically not know the VPN units' static private key, $S_i^{st}(t)$, nor will it know their dynamic keys. The KRA has static private and public keys associated with it, S_{KRA}^{st} and P_{KRA}^{st}.

Common keys between two entities (two VPN units or one VPN unit and its KRA) are always calculated by each party. These common keys are obtained by each node or KRA performing functions on the other node's or KRA's public key. As a result, the

common keys are never transmitted and consequently the common keys represent a shared secret between the two entities.

Normal operation

Each VPN unit and a corresponding KRA negotiate a static common session key, $K_{KRA,i}^{st}$, using, e.g., the Diffie-Hellman key exchange protocol (exchanging their static public keys). After exchanging their public keys, a single common session key is calculated by both sides from their own static private keys and the other party's static public key. As previously mentioned, the KRA also stores the static public key of all VPN units with which it performed a Diffie-Hellman key exchange. Depending on the need, these public keys may be released to third parties (such as government agencies), as desired.

In regard to exchanges between two VPNs (which each may be hardware, software or a combination thereof), units A and B also illustratively use the Diffie-Hellman key exchange protocol (exchanging their static public keys) to develop their static common crypto key, $K_{A,B}^{st}$. Note that this key will not be used to encrypt or decrypt messages but instead will be used in the dynamic public key exchange.

Once the static common key is calculated, units A and B perform a second Diffie-Hellman key exchange protocol. In the second exchange, each unit A and B with exchange their respective dynamic public keys encrypted with the static common key, $K_{A,B}^{st}(t)$, previously calculated. Based on the received dynamic public keys encrypted with the other units static common key, a dynamic common key $K_{A,B}^{dyn}$ is calculated.

Note that during the second Diffie-Hellman exchange, when unit A sends its dynamic public key to unit B (encrypted with the static common key shared by A and B), it attaches its dynamic private key encrypted with its common session key shared with the KRA. A

time stamp is also attached. Illustratively, the message transmitted by unit A to unit B comprises:

$$E_{K^{st}_{A,B}}(t)(P_A^{dyn}(t)),E_{K(KRA,A)}(S_A^{dyn}(t)),t$$

From this message, unit B can decrypt unit A's dynamic public key $P_A^{dyn}(t)$. However, since unit B does not know the static common key shared by KRA and unit A, unit B can not decrypt unit A's dynamic private key. Unit B will also send unit A its dynamic private key encrypted with the common key it shares with its KRA, along with a time stamp.

Listening by an "Authorized" 3rd party

To decrypt messages by a third party, the following steps are followed, which are to be accompanied by the flowchart of FIG 3.

In step 50, the Authority which is, e.g., authorized by a Court Order, starts recording the decrypted messages between units A and B. In step 55, the Authority retrieves the static public key of unit A, P_a^{st}, and the static private key of KRA, S_{KRA}^{st}, from KRA and from these it calculates the static common session key between KRA and unit A, $K^{st}_{KRA,A}$.

Next, in step 60, the Authority retrieves the second D-H exchange message from A→B, $E_{K(KRA,A)}(S_A^{dyn}(T))$, and after decryption, it obtains the dynamic private key of unit A, $(S_A^{dyn}(T))$. The Authority then retrieves the second D-H exchange message from B→A, $E_{K(KRA,B)}(S_B^{dyn}(T))$, and after decryption, it obtains dynamic private key of unit B $(S_b^{dyn}(T))$, in step 65. Then, in step 70, the Authority calculates the dynamic public key of unit B, $P_B^{dyn}(T)$, from $S_b^{dyn}(T)$.

Lastly, the Authority calculates the dynamic common session key of units A and B, $K^{dyn}_{A,B}(T)$, from $S_A^{dyn}(T)$ and P_B^{dyn}, in step 75 The dynamic common session key is the key needed to decrypt the messages in question between units A and B.

Conclusion

The inventive method of the present invention may be summarized by the following steps below:

1. Nodes I, I=1, N are communicating with each other encrypted. They each have static private (S_i) and public (P_i) keys, which never change and dynamic private (S_i^{dyn}) and public (P_i^{dyn}) keys, which are functions of time (t).
2. The Key Recovery Authority (KRA) also has static private (S_{KRA}) and public (P_{KRA}) keys, which never change. The KRA exchanges static public keys with each of the nodes, thus develops a static common key (session key), $K_{KRA,i}$, with each of them using, for example, the Diffie-Hellman protocol.
3. The KRA maintains a list of the static public keys of all nodes. Thus, the (static) session key with any of the nodes can be "recovered" at any time.
4. When two nodes, say i and j, exchange their dynamic public keys (encrypted with their static session key $K^{st}_{ij}(t)$), then each one attaches its dynamic secret key, encrypted with the static session key between it and the KRA. A time stamp is also included:

$$EK^{st}_{ij}(t)\,(P_i^{dyn}(t))\,E_{K(KRA,i)}\,(S_i^{dyn}(t)),\,t$$

5. With the knowledge of the session key, $K_{KRA,i}$, which can be recovered from the KRA (as described in steps 2 and 3), the dynamic private keys of each node, $S_i^{dyn}(t)$, can be recovered (and $P_i^{dyn}(t)$ calculated) from a recording of any session. From

$S_i^{dyn}(t)$ and $P_j^{dyn}(t)$ one can calculate the dynamic session key between the two nodes ($K_{ij}^{dyn}(t)$). However, all other parties are still protected since their dynamic public keys are exchanged encrypted.

6. All nodes are still protected, and their session concealed, because their private keys are encrypted.

The above described embodiments of the invention are intended to be illustrative only. Numerous alternative embodiments may be devised by those skilled in the art without departing from the spirit and scope of the following claims.

CLAIMS

What is claimed is:

1. A method of determining a dynamic common key for decrypting messages transmitted between first and second nodes by a third party, comprising the steps of:

retrieving a static public key of the first node, P_A^{st}, and a static private key of a corresponding key recovery authority (KRA) node, S_{KRA}^{st}, from said KRA node, wherein said KRA node has a static public key of each of said first and second nodes stored therein;

determining a static common session key, $K^{st}_{KRA,A}$, between said KRA and said first nodes, based on said P_A^{st} and S_{KRA}^{st};

retrieving a first exchange message, $E_{K(KRA,A)}(S_A^{dyn}(T))$, transmitted from said first node to said second node;

determining a dynamic private key of said first node, $(S_A^{dyn}(T))$, based on said $E_{K(KRA,A)}(S_A^{dyn}(T))$;

retrieving a second exchange message, $E_{K(KRA,B)}(S_B^{dyn}(T))$, transmitted from said second node to said first node;

determining a dynamic private key of said second node, $(S_B^{dyn}(T))$, based on said $E_{K(KRA,B)}(S_B^{dyn}(T))$;

determining a dynamic public key of said second node, $P_B^{dyn}(T)$, based on said $S_B^{dyn}(T)$; and

determining said dynamic common key, $K^{dyn}_{A,B}(T)$, based on said $S_A^{dyn}(T)$ and said P_B^{dyn}, for decrypting messages transmitted between said first and second nodes by said third party.

2. The method of claim 1, wherein said first and second exchange messages include a time stamp.

3. The method of claim 1, wherein said first and second nodes comprise respective cryptography devices.

4. A method of decrypting a dynamic public key of a first node by a second node, comprising the steps of:

retrieving a static public key, P_A^{st}, from said first node;

determining a static common key, $K_{A,B}^{st}$, based on said P_A^{st};

retrieving a dynamic public key, P_A^{dyn}, from said first node which is encrypted with said $K_{A,B}^{st}$; and

retrieving a dynamic private key, S_a^{dyn}, from said first node which is encrypted with a common session key between said first node and a key recovery authority (KRA) third party node $K_{KRA,A}$;

wherein said S_a^{dyn} encrypted with said $K_{KRA,A}$ is utilized for decrypting said dynamic public key of said first node.

5. The method of claim 4, wherein said step of determining said static common key, $K_{A,B}^{st}$, is further based on a static private key of said second node.

6. The method of claim 5, wherein said S_a^{dyn} encrypted with said $K_{KRA,A}$ further includes a time stamp.

7. The method of claim 4, wherein said first and second nodes comprise respective cryptography devices.

8. A transmitted data message, transmitted from a first node to a second node, for decrypting the first node's dynamic public key, comprising:

a dynamic public key, P_A^{dyn}, from said first node which is encrypted with a static common key between said first and second nodes, $K_{A,B}^{st}$; and

a dynamic private key, S_a^{dyn}, from said first node which is encrypted with a common session key between said first node and a key recovery authority (KRA) third party node $K_{KRA,A}$,

wherein said S_a^{dyn} encrypted with said $K_{KRA,A}$ is utilized for decrypting said dynamic public key of said first node.

9. The message of claim 8, wherein said first node comprises a cryptography device and said second node is a key recovery authority (KRA) third party node.

10. The message of claim 8, wherein said first and second nodes comprise respective cryptography devices.

11. The message of claim 8 further comprising a time stamp.

12. A Key Recovery Authority (KRA) device for calculating a dynamic common key between a first and a second node for decrypting messages transmitted between said first and second nodes, said KRA device comprising:

a memory for storing a respective static public key of each of said first and second nodes; and

a processor for:

retrieving said static public key of the first node, P_A^{st}, with a static private key of said KRA device,

determining a static common session key, $K^{st}_{KRA,A}$, between said KRA device and said first node, based on said P_A^{st} and S_{KRA}^{st},

retrieving a first exchange message, $E_{K(KRA,A)}(S_A^{dyn}(T))$, transmitted from said first node to said second node,

determining a dynamic private key of said first node, $(S_A^{dyn}(T))$, based on said $E_{K(KRA,A)}(S_A^{dyn}(T))$,

retrieving a second exchange message, $E_{K(KRA,B)}(S_B^{dyn}(T))$, transmitted from said second node to said first node,

determining a dynamic private key of said second node, $(S_B^{dyn}(T))$, based on said $E_{K(KRA,B)}(S_B^{dyn}(T))$,

determining a dynamic public key of said second node, $P_B^{dyn}(T)$, based on said $S_B^{dyn}(T)$, and

determining said dynamic common key, $K^{dyn}_{A,B}(T)$, based on said $S_A^{dyn}(T)$ and said P_B^{dyn}, for decrypting messages transmitted between said first and second nodes by said third party.

13. The KRA device of claim 12, wherein said first and second exchange messages include a time stamp.

14. The KRA device of claim 12, wherein said first and second nodes comprise respective cryptography devices.

Fig. 1A (Prior Art)

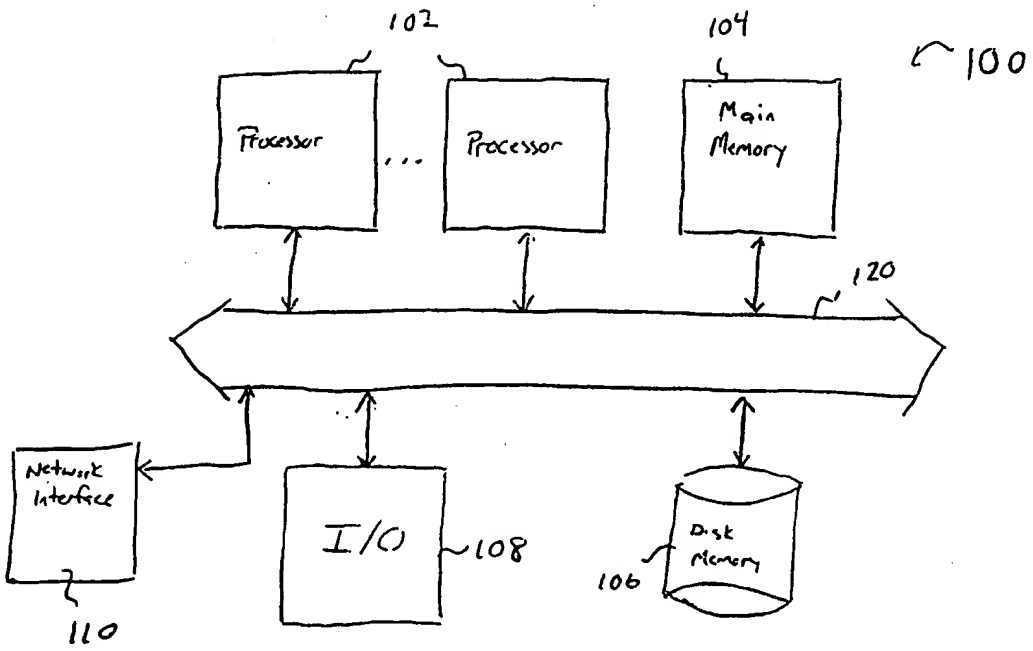


Fig. 1B (Prior Art)

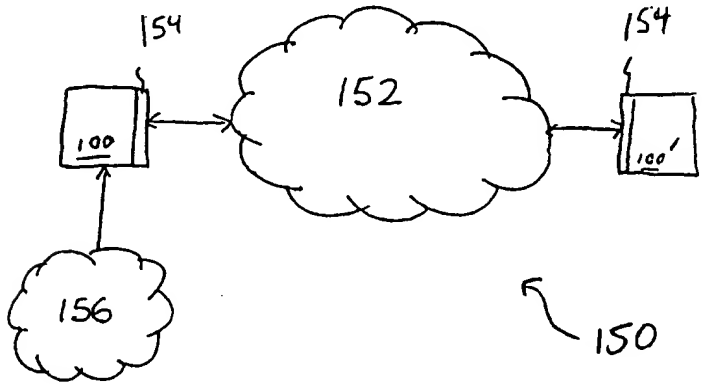
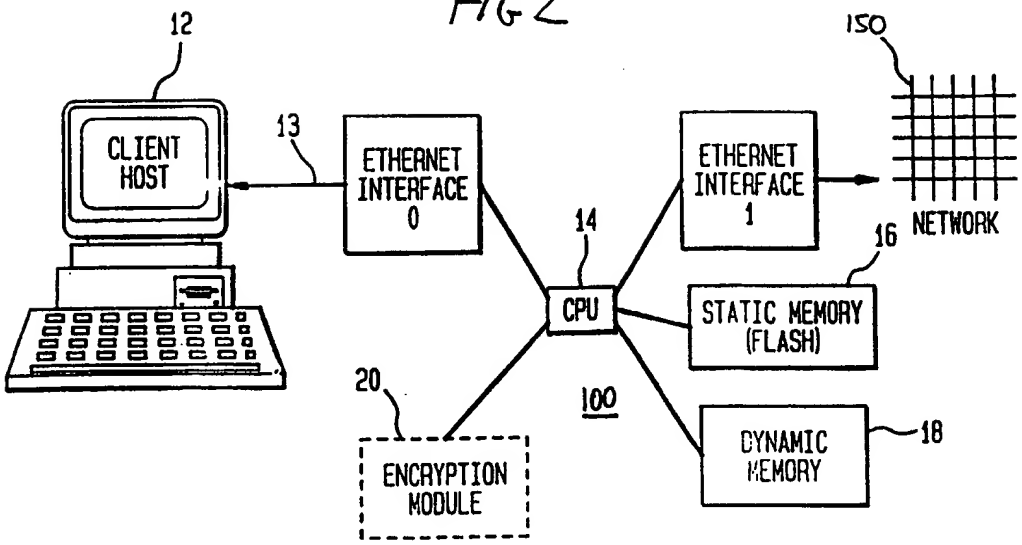


FIG 2


12
CLIENT HOST
13
ETHERNET INTERFACE 0
14
CPU
ETHERNET INTERFACE 1
150
NETWORK
16
STATIC MEMORY (FLASH)
100
DYNAMIC MEMORY
18
20
ENCRYPTION MODULE

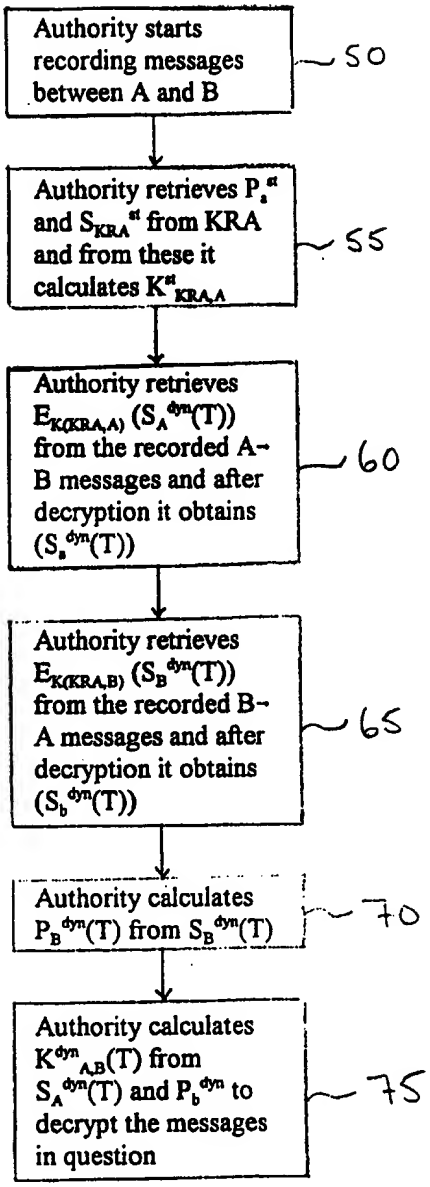

Authority starts recording messages between A and B
50
Authority retrieves P_a^{st} and S_{KRA}^{st} from KRA and from these it calculates $K^{st}_{KRA,A}$
55
Authority retrieves $E_{K(KRA,A)}$ ($S_A^{dyn}(T)$) from the recorded A→B messages and after decryption it obtains ($S_a^{dyn}(T)$)
60
Authority retrieves $E_{K(KRA,B)}$ ($S_B^{dyn}(T)$) from the recorded B→A messages and after decryption it obtains ($S_b^{dyn}(T)$)
65
Authority calculates $P_B^{dyn}(T)$ from $S_B^{dyn}(T)$
70
Authority calculates $K^{dyn}_{A,B}(T)$ from $S_A^{dyn}(T)$ and P_b^{dyn} to decrypt the messages in question
75

FIG. 3

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US99/03665

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :H04L 9/08
US CL :380/21

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 380/21, 23, 25, 30, 49

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

APS (search terms: key recovery, key escrow, trusted authority, Law Enforcement Access Field, (static(3a)key# and dynamic(3a)key#), trusted third party)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5,557,346 A (LIPNER et al.) 17 September 1996, see summary and background	1-14
A	US 5,557,765 A (LIPNER et al.) 17 September 1996, see summary and background	1-14
A	US 5,631,961 A (MILLS et al.) 20 May 1997, see summary and background	1-14
A	US 5,633,929 A (KALISKI, JR.) 27 MAY 1997, see background and summary	1-14
A	US 5,640,454 A (LIPNER et al.) 17 June 1997, see background and summary	1-14

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

*	Special categories of cited documents.	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search 17 AUGUST 1999	Date of mailing of the international search report 10 SEP 1999
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer Pinchus Laufer James R. Matthews Telephone No. (703) 306-5539

Form PCT/ISA/210 (second sheet)(July 1992)*